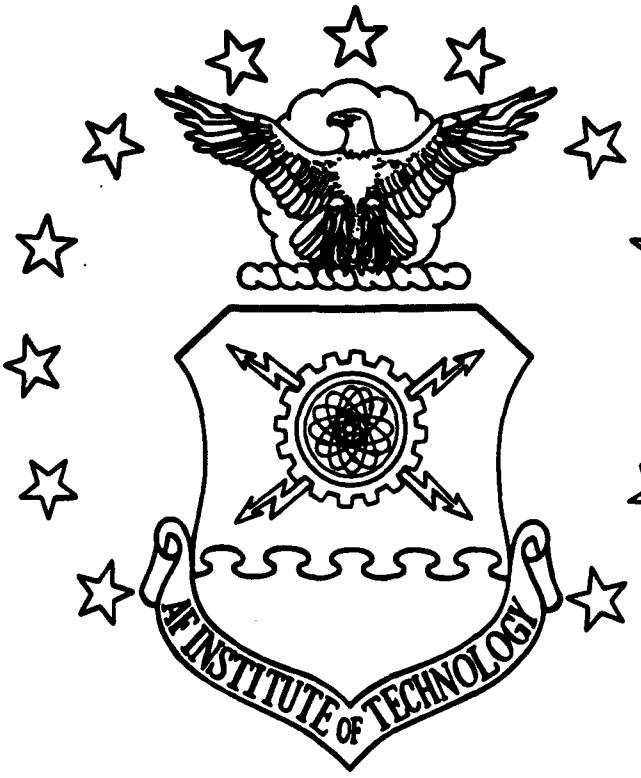

INVESTIGATIVE SEARCH OF QUALITY HISTORICAL SOFTWARE SUPPORT COST DATA AND SOFTWARE SUPPORT COST-RELATED DATA

THESIS

Brent L. Barber, Captain, USAF

AFIT/GSS/LSY/91D-1

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GSS/LSY/91D-1

INVESTIGATIVE SEARCH OF QUALITY HISTORICAL SOFTWARE SUPPORT COST DATA AND SOFTWARE SUPPORT COST-RELATED DATA

THESIS

Brent L. Barber, Captain, USAF

AFIT/GSS/LSY/91D-1

The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

Accesion For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

AFIT/GSS/LSY/91D-1

INVESTIGATIVE SEARCH OF QUALITY HISTORICAL SOFTWARE SUPPORT COST DATA AND SOFTWARE SUPPORT COST-RELATED DATA

THESIS

Presented to the Faculty of the School of Systems and Logistics of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Master of Science in Software Systems Management

Brent L. Barber, B.S.

Captain, USAF

December 1991

Preface

The purpose of the study was to determine what software support cost data and software support cost-related data should and can be collected by the Air Force. In addition, the study was to determine how the subject data should be collected.

A review of the Air Force "preferred" software support cost models was performed to determine the inputs required by the models and the outputs received from the models. In addition, several telephone interviews were conducted with personnel knowledgeable in the area, to determine the current status of data collection. Based on the information gathered, recommendations were made.

The collection of this data is necessary in order to determine the accuracy and calibration requirements of the software support cost models used by the Air Force. The Air Force must be aware of the costs involved with supporting the weapon systems that are being put into operational use.

Many individuals contributed significantly in the successful completion of this thesis effort. Without the patience, advice, and encouragement of my thesis advisor, this would have been a far more difficult task. I would like to thank my family for their continuing support. They were always there when I needed them. Finally, I would like to thank my friends Tom, Sean and Julie for always helping me keep things in perspective.

Brent L. Barber

Table of Contents

List of Tables

Abstract

The purpose of the study was to determine what software support cost data and software support cost-related data should and can be collected by the Air Force. In addition, the study was to determine how the subject data should be collected. The collection of this data is necessary in order to measure the accuracy and calibration requirements of the software support cost models used by the Air Force.

The study found there was no standard set of data items, and no standard procedures in place for the collection of this data. In most cases data were available or easily accessible, but the data had not been requested, therefore, it was not collected. A few organizations were collecting data, but mainly for the purpose of reporting to their superiors.

Data were obtained by reviewing the Air Force "preferred" software support cost models and through semi-structured telephone interviews with individuals knowledgeable in the area.

Recommendations to improve the data collection effort were provided. Among the recommendations was the use of a standard data set and standard collection procedures. The study includes a proposed data collection form containing all items suggested for inclusion in the standard data set.

INVESTIGATIVE SEARCH OF QUALITY HISTORICAL SOFTWARE SUPPORT COST DATA AND SOFTWARE SUPPORT COST-RELATED DATA

I. Overview

Introduction

The role of software in Air Force and Department of Defense (DoD) weapon systems is increasing steadily every year. Software plays a major part in the majority of DoD's mission critical assets. These assets include aircraft, missiles, ships, submarines, tanks, satellites, and command and control systems. Major improvements in computer hardware have created a need for larger, more complex software programs and, as the role of software expands, the cost increases (3:10).

"The Air Force spent about $4 billion on software in 1987-roughly five percent of the total Air Force budget.", stated Lt Gen James S. Cassity, Jr., the former Commander of Air Force Communications Command. Although software costs are already significant, most believe these costs will continue to grow. Mr. Lloyd Moseman, the former Air Force Deputy Assistant Secretary (Logistics), has predicted software development and maintenance costs could be as much as $30 billion by 1995 (21:81).

An extremely large portion of a program's software life cycle costs, frequently 50 to 75 percent, are for software

support or, as it is sometimes called, software maintenance (2:533). Software support includes all software activities after the product has been delivered to the customer.

With software costs and the future requirements for software increasing yearly, one of the major concerns is that the DoD has not adequately estimated or tracked software support costs. As reported in the DoD Software Master Plan, better visibility of software support costs is essential (9:7). Without accurate support cost data, projecting resource requirements on a long-term basis is very difficult (22:25). The Air Force Cost Center has identified four software support cost estimating models as the "preferred" Air Force models; but, at this time, no large-scale quantitative study has been performed. This type of study is necessary in order to measure the accuracy and calibration requirements of the models (11). The primary reason no studies have been performed for software support cost models, is the scarcity of quality historical software support cost data (5:4). Without this quantitative and qualitative information, no direct comparison between estimates and actual data can be performed; therefore, the accuracy of the models is still questionable (6:4). By collecting historical data, this comparison can be accomplished. Even if models are not completely accurate, they can be calibrated to reduce the error.

Specific Research Problem

Given the requirement for both quality historical software support cost data, and quality software support cost-related data, what data should and can be collected, and how should the data be collected?

Investigative Questions

The following are the investigative questions that will be answered by this study:

1. What type of software support cost estimates (outputs) are provided by the Air Force Cost Center "preferred" software support cost estimating models?

2. What type of software support data (inputs) are required for the Air Force Cost Center "preferred" software support cost estimating models?

3. What types of mission critical software support cost data and software support cost-related data are currently collected or can be collected by the Air Force?

4. What are the current procedures used by Air Force organizations for gathering historical mission critical software support cost data and software support cost-related data?

5. In the opinion of knowledgeable personnel in the area, what historical mission critical software support cost data and software support cost-related data should be gathered, and how should these data be gathered?

Definition of Terms

Mission Critical Computer Resources (MCCR) Software. "...includes the following five applications along with their respective sub-applications:

1) Intelligence activities
2) Cryptological activities for national security
3) Command and control
4) Equipment integral to a weapon system (i.e.., Embedded Computer Resources)
5) Resources critical to military and intelligence missions" (14:2-2).

Post-Deployment Software Support. "Those software support activities that occur during the full-rate production and operations support phases of the acquisition process" (8:8).

Software. "A combination of associated computer instructions and computer data definitions required to enable the computer hardware to perform computational or control functions" (10:20).

Software Development Process. An eight phase process consisting of:

1. System Requirements Analysis
2. Computer Software Configuration Item Analysis
3. Preliminary Design
4. Detailed Design
5. Coding and Computer Software Unit Testing
6. Computer Software Component Integration and Test
7. Computer Software Configuration Item Testing
8. System Integration and Testing

(7:9).

Software Maintenance. "Changes that have to be made to computer programs after they have been delivered to the customer or user" (14:16-1).

Software Support. "In the Air Force, software support includes all software activities after the software is developed and the first production systems or items are given to the Using Command for operational use" (15:7-4). For the remainder of this document, the terms software support, post-deployment software support and software maintenance will be synonymous.

Scope

The scope of this research project is limited to identifying what software support cost data can and should be collected by the Air Force, and how the data should be collected. This study will be limited to Air Force (MCCR) software support costs. Costs incurred during the software development process will not be examined. In addition, Air Force automatic data processing equipment software, such as payroll systems software, will not be included.

Summary

Both the importance and cost of software is increasing annually, with a major portion of the costs resulting from software support. As these costs rise, so does the importance of software support cost models that can adequately estimate and track costs. For these models to be

effective, it is essential that quality historical cost data, and cost-related data, be available to measure the accuracy and to calibrate software support cost estimating models.

II. Background

Introduction

This chapter is organized into three basic sections. The first two sections provide background on the software development process and post-deployment software support process, and the third section discusses software support cost estimating techniques and models. This chapter examines a sample of published technical reports, books and periodicals concerning the topics mentioned above.

Software Development Process

Various descriptions of the software development process were identified in the literature (2:35-54;10:25;14:3-1 to 3-9). The software development process described in this chapter is defined as is shown in the definition of terms provided in Chapter 1. The software development process comprises eight phases:

1. System Requirements Analysis
2. Software Requirements Analysis
3. Preliminary Design
4. Detailed Design
5. Coding and Computer Software Unit Testing
6. Computer Software Component Integration and Testing
7. Computer Software Configuration Item (CSCI) Testing
8. System Integration and Testing (7:9)

These phases are found in DoD-STD-2167A, which is the DoD standard for MCCR software development (7:9).

System requirements analysis/design begins the software development process. The purpose of this phase is to define and analyze the requirements of the system to ensure the software requirements are consistent and complete. This analysis will determine the optimal allocation of hardware, software and personnel, so the system may be partitioned into hardware configuration items (HWCIs) and computer software configuration items (CSCIs). A preliminary set of engineering requirements for each of the CSCIs and the preliminary set of interface requirements are identified during this phase.

The next phase of the process is software requirements analysis. During this phase, a complete set of engineering requirements and interface requirements are identified (7:9).

Once the requirements for the software are defined, the preliminary design phase begins. During this phase, an initial design is developed and trade off studies are performed.

The next phase in the software life cycle is detailed design. Refinement of the preliminary design is the purpose of this phase.

Completion of the detailed design phase signals the start of the coding and unit testing phase. This phase, as the title suggests, consists of coding the design, and

testing the individual units to ensure each unit performs its intended function.

Component integration and testing is the sixth phase of the life cycle. The purpose of this phase is to ensure proper interaction and integration.

The next phase of the life cycle is CSCI testing. During this phase, the complete CSCI is tested to ensure all requirements identified during the requirements analysis phase are met (14:3-1 to 3-9).

The final phase is system integration and test which consists of the integration and testing of all the CSCIs in the system to ensure all system requirements are met (7:34). Although this is the last phase of the software development cycle, it is likely the software configuration item will undergo further changes, which will result in many iterations of part or all of the software development cycle.

Post-Deployment Software Support

Post-deployment software support (PDSS) or, as it is sometimes called, software maintenance, occurs when changes are made to software after the user has taken delivery (20:3). Ferens suggests the term that may be most appropriate for this activity is "redevelopment" (16:2). There are numerous reasons for changing software and these changes typically fall into three different categories: corrective, adaptive, and perfective (18:13-14). Corrective maintenance is performed to correct errors in software

performance or implementation (20:22). According to Glass and Noiseux, these changes make up approximately seventeen percent of the overall software support activity (18:14). The next category is adaptive maintenance, which consists of changes required in order to interface with external environments that have changed (20:22). These types of software changes account for approximately eighteen percent of the software support activity. The final type of software support is perfective maintenance. Perfective maintenance encompasses changes which improve cost-effectiveness, processing efficiency or maintainability (20:22). These changes usually take place as a result of a user's request to extend the life of the software past its initial requirements (16:2). Approximately sixty percent of software support activity falls into this category according to Glass and Noiseux (18:13). The remaining five percent, not included in the three major categories is classified as "other" maintenance by Glass and Noiseux (18:14).

The Air Force Logistics Command (AFLC) compiled information on software support activity and partitioned the data into seven categories; this data is provided in Table 1. By equating correction of design errors to corrective maintenance, responding to threat changes to adaptive maintenance, and the remaining categories shown in Table 1 to perfective maintenance, the results of the AFLC study are quite similar to the figures provided by Glass & Noiseux (14:16-2).

TABLE 1

AFLC SUPPORT ACTIVITIES (14:16-2)

Activity	Percentage
CORRECT DESIGN ERRORS	17
RESPOND TO THREAT CHANGES	21
ADD NEW CAPABILITIES	27
DELETE UNNEEDED CAPABILITIES	6
IMPROVE EFFICIENCY OR EFFECTIVENESS	10
INTERFACE WITH OTHER SYSTEMS	12
IMPROVE HUMAN FACTORS OR RELIABILITY AND MAINTAINABILITY	7

Each time a software change is required, all or some part of the software development life cycle will be repeated. As stated in Chapter 1, these changes can be quite costly, many times exceeding the cost of initial development (2:533). Further evidence of this is provided by a survey conducted by Lientz and Swanson which discovered that, in the large organizations they surveyed, approximately fifty percent of programming time was allocated to maintain existing software (24:534). These figures show the significance of software maintenance cost in the total cost of software.

Post-Deployment Software Support Process

When changes are made to software following deployment, the changes will be incorporated through the post-deployment software support process. According to MIL-HDBK-347, this process consists of four general phases: initial analysis, software CSCI development, system integration and testing, and product logistics. The PDSS process is typically initiated when the user perceives a problem with the system and a problem/change report is submitted to the logistics support organization, the program office, or the software support agency (SSA). The first phase of the PDSS process is the initial analysis phase. This phase determines the impact of the software change, the estimated effort required for the change, and the risks associated with its implementation. The next phase is the software CSCI development phase, which involves isolation, implementation and testing the software change. Upon completion of software development, the CSCIs must be integrated and tested to identify and correct any faults. This phase is referred to as the system integration and testing phase. When all integration and testing is complete it is time for the product logistics phase. During this phase, the logistics to support the new configuration must be put into place (8:25-27).

Software Cost Estimating

Software project costs can be broken down into three components: hardware costs, travel and training costs, and effort costs. Effort costs, the costs an organization spends for personnel during a software project, make up the largest portion of project costs and are the most difficult to estimate. For this reason, and because software managers often require project cost estimates, many cost estimating techniques have been developed (24:514).

According to Ferens, software cost estimation basically falls into four categories:

1. Analogy
2. Expert Judgement
3. Bottom-Up
4. Top-Down

Each of these types has advantages and disadvantages.

The simplest type of model is the analogy model, which compares project costs of similar projects to the project in question. These models can be quite effective because the estimate provided is based on experience. The major disadvantage of this type of model is that often similar types of projects do not exist or cannot be identified. This type of model is not commonly used by DoD because in most cases similar projects cannot be found.

Expert judgement cost estimating models utilize the knowledge and experience of one or more experts to estimate the cost of a project. This type of model can be effective

in some situations, but often experts cannot be identified. Even if experts can be located, the estimates can be subject to the personal biases of the experts. Another concern with this method is that the actual knowledge of experts can be insufficient. Again, this type of model is not commonly used by DoD.

A more detailed type of model is the bottom-up approach. Bottom-up techniques estimate costs by breaking down the project into components or units and then summing the costs. Each unit in the system has an extensive cost analysis performed on it. Because a detailed analysis is performed, these types of models are typically more stable and accurate than other models. In addition, because projects are broken down into smaller units, individuals or small groups may take responsibility for keeping costs below those estimated. One of the disadvantages of this type of model is that integration costs may not be accounted for. Additionally, the process is very time consuming and detailed information, which is often not available, especially in the early phases of a project, is required. For these reasons, these types of models are seldom used by DoD.

The final type of technique is the top-down approach, which is sometimes called parametric estimating. This method estimates software costs by examining design characteristics, or parameters, of software programs. Top-down models are fast, easy to use, and typically do not

require much detailed information. The major disadvantage of these types of models is their lack of accuracy. Due to the ease of use and to the timeliness of these models, these models are the most widely used in DoD (14:10-1 to 10-3), and are the ones considered in this study.

"Preferred" Software Support Cost Estimating Models

The Air Force Cost Center, the focal point for software cost estimating models has identified four software cost estimating models as the "preferred" models for Air Force usage. These models are: The Revised Enhanced Version of Intermediate COCOMO (REVIC) model, the Software Architecture, Sizing, And Estimating Tool (SASET), the System Evaluation and Estimation of Resources (SEER) model, and the PRICE Software Model (PRICE-S) (11).

The REVIC, the first of the "preferred models, was originally developed for the Air Force by an Air Force reservist, Raymond Kile. This model is currently maintained and managed by the Air Force Cost Center. The acronym REVIC initially stood for Ray's Enhanced Version of Intermediate COCOMO, but has since been changed to Revised Enhanced Version of Intermediate COCOMO. As the acronym suggests, the REVIC model is based on the COnstructive COst MOdel for Maintenance (COCOMO-M), which was developed by Dr. Barry Boehm of TRW. COCOMO-M was developed by performing analysis on sixty-three past software projects. The model is an extension of COCOMO which estimates software development

costs (16:3). Like COCOMO and COCOMO-M, REVIC utilizes information concerning the attributes of the software product, the computer, the personnel working the program, and attributes of the project to estimate both development and maintenance costs. REVIC computes fifteen years of maintenance costs. The major differences between REVIC and COCOMO are changes in the coefficients in the basic equation, and the addition of a separate mode for the Ada programming language. The REVIC mode typically provides a higher estimate than COCOMO due to the higher coefficients in the basic equation. The Air Force's Contract Management Division (AFCMD) validated REVIC for development costs by using data provided by the Air Force's Rome Air Development Center (RADC). The validation study did not use the same data base which was used in the initial calibration effort. The REVIC model is available to DOD personnel in a user-friendly personal computer (PC) version (19:4,7).

The SEER model is the second of the Air Force Cost Center's "preferred" models. This model is the property of Galorath Associates, Incorporated, of Marina del Rey, California, and is based on the work of Dr. Randall Jensen (16:4). The model is based on the hypothesis of P.V. Norden who believed software development manpower efforts follow a Rayleigh distribution function for learning (4:4). The model provides both development and maintenance costs and has a separate mode for support costs and support-unique inputs (17:35-36). Calibration of the model is very

difficult, but can be accomplished by adjusting the technology factors and by changing inputs (13:C9-4). SEER is a proprietary model and can be leased on an annual basis in its PC version (16:4).

The third "preferred" software support cost estimating model is the General Electric (GE) PRICE (PRICE-S) model. The PRICE-S model was developed by Mr. Frank Freiman, using expert opinions. It was first made commercially available by RCA-PRICE Systems Division in 1977. This model also estimates development and maintenance costs. The model provides a separate mode for maintenance, which estimates support costs based on the development inputs and support-unique inputs. As with the REVIC model; product, computer, personnel and project attributes are used by PRICE-S to produce an estimate (4:7-8). PRICE-S does have a user-friendly calibration option (13:C9-4). The model does not currently have a PC version, but it is available from a time sharing service (16:3).

The Software Architecture, Sizing and Estimating Tool (SASET) is the fourth Air Force Cost Center "preferred" model. It was developed by Martin Marietta of Denver, Colorado for DoD. This model is managed and maintained by the Naval Center for Cost Analysis and the Air Force Cost Center. SASET accepts seventeen inputs in its "maintenance" tier, and uses both support and developmental parameters to estimate support costs. This model is limited to use by DoD personnel and is available only in a PC version (23:2-3).

Other Software Support Cost Estimating Models

The Avionics Software Support Cost Model (ASSCM) was developed by SYSCON Corporation in 1983 for use by the Air Force Wright Aeronautical Laboratories (AFWAL). The model estimates only software support costs, unlike the other models that are covered, which provide estimates for both software development and software support costs. The model is based on historical data from the F-111F, F-16 and A-7 Air Force programs. ASSCM uses twenty-three input variables including both support specific and developmental parameters (25:9-12). The model is not proprietary and is currently available only in VAX format, which has been a major drawback to its use (16:3).

Checkpoint is based on the work of Capers Jones, and is owned by Software Productivity Research of Cambridge, Massachusetts. This model estimates development and maintenance costs as well as software reliability and quality. Calibration of Checkpoint can only be accomplished by changing inputs. This model is proprietary and is available in a PC version (13:C9-4).

The Software Life Cycle Model (SLIM) is owned and maintained by Quantitative Software Management, Incorporated, of McLean Virginia. The developer of this model is Lawrence H. Putnam. Like SEER, SLIM is based on the Rayleigh-Norden curve and estimates both development using several development inputs and the 60/40 ratio of support to development. A calibration option is provided by

the model (4:4). SLIM is also proprietary and is available for lease on a yearly basis in a PC version (16:4).

The final model that discussed is the System-4 Model. This model is based on the work of Dr. Randall Jensen with some modifications made by Computer Economics Incorporated, of Marina del Rey, California. The model utilizes developmental parameters to determine both development and maintenance cost estimates. Like many of the other models, calibration of the model is possible. This proprietary model can be leased on an annual basis in a personal computer version (16:3-5).

These models, like the "preferred" models, have support cost capability but, because of the limited time available for a thesis, and because they were not currently one of the Air Force Cost Center's "preferred" models, they were not analyzed.

Each of the cost models discussed has something in common. They all require information from the using organization, and no one model has been proven to accurately estimate costs in all situations (14:10-2).

III. Methodology

Introduction

This chapter describes the methods that were used to answer the investigative questions in Chapter I. The data sources, data collection process, and data analysis procedures are described.

Data Sources

Data were obtained via a comprehensive literature review, an extensive review of the "preferred" software support cost models, and telephone interviews with DoD personnel knowledgeable in the area of software support. Respondents were selected based upon their positions relative to the research subject. Personnel were identified through conversation with members of the AFIT faculty and students, exploratory interviews, and interviews with respondents. A semi-structured interview was developed to allow respondents to freely express their views and yet provide the required information. The list of questions for the interview was developed based on data found in the literature review, and the review of the "preferred" cost models. The questionnaire was reviewed by an AFIT faculty member and a member of the Air Force Cost Center. The interview was designed to last approximately twenty to thirty minutes, which is slightly longer than the amount of

time that is typically required for this type of interview (12:171).

The telephone interview was considered to be the most appropriate type of instrument. Telephone interviews provide an expedient method to collect a sufficient amount of information from a large number of individuals. Telephone interviews are lower in cost, have a higher response rate, and do not require travel. In addition, according to Emory, it is likely that interview bias can be reduced by performing telephone interviews versus personal interviews (12:169-170).

Data Collection

The following procedures were used to gather data for this research effort:

Step 1. Review of the literature to determine what input data are required and what output data are currently provided by the various software support cost estimating models used by the Air Force.

Step 2. Review literature to determine what historical software support cost data, and software support cost-related data, are collected by the Air Force.

Step 3. Review literature to determine what procedures are in place to collect historical software support cost data, and software support cost-related data, by the Air Force.

Step 4. Perform telephone interviews with personnel knowledgeable in the area to determine: what historical software support cost data, and software support cost-related data, are being collected by the Air Force, what data can be collected, how data are currently collected, what historical data should be collected, and how data should be collected. The following procedures were used to gather the telephone interview data:

1. The respondents (Appendix A) were contacted by telephone to establish a rapport, explain the purpose of the study and to discuss the nature of the questionnaire.

2. Following the initial contact, a letter listing the dates of the interviews was sent (Appendix B). The letter included a copy of the questionnaire (Appendix C), and a return telephone number. The telephone number provided potential respondents the opportunity to clarify any questions concerning the questionnaire or make arrangements to conduct the telephone interview either earlier or later than the listed dates.

3. A follow-up phone call was made to schedule a specific date and time for the interview, and to ensure all respondents had received a copy of the questionnaire. This phone call also provided another means to answer any questions concerning the questionnaire. At the time of this contact, each respondent was informed his/her responses would remain anonymous. Anonymity was chosen to encourage candid answers to the questions.

4. The interviews with the respondents were conducted as scheduled.

5. Follow-up interviews were performed with some respondents to clarify responses.

6. A summary of the verbal responses to each of the questions was entered into a spreadsheet and hard copies of the data were printed to ease data analysis.

7. A letter of appreciation (Appendix D), and a copy of the approved thesis were sent to each of the respondents. (All respondents expressed a desire to receive a copy of the study.)

Data Analysis

The data obtained through the literature review, the review of the "preferred" models, and the telephone interviews were analyzed to determine what software support cost data and software support cost-related data should and could be collected, and how the data should be collected. The data were compared and contrasted using a matrix of the cost model data inputs and outputs and responses from the respondents. The matrix was developed to assist in the data analysis process. The data provided sufficient information to answer the investigative questions and the specific research question in Chapter I.

Significant Hurdle

The most significant hurdle was reducing bias in the personal interview portion of the research process. To reduce bias, potential respondents were initially contacted by telephone and a format sheet was used to explain the purpose of the study, and to establish a rapport. Following the initial contact, an introductory letter was sent providing the dates of the interviews, and included a copy of the interview questions. This procedure allowed respondents an opportunity to prepare for the interview and provide them a point of contact in the event clarification or rescheduling of the interview was required.

IV. Findings and Data Analysis

Preface

Data were collected by reviewing literature, examining the Air Force Cost Center "preferred" software support cost estimating models, and by interviewing nine individuals knowledgeable in Air Force post-deployment software support. A general description of each respondent's job position and duty station is listed in Appendix A.

All data were used to answer the investigative questions and the specific research question. The initial step in the analysis was the development of a list of data items which represented the inputs and outputs of the "preferred" models. This list was used in the questionnaire to the respondents to determine which data items are, or can be, collected. The data obtained from the review of the models and the interviews were analyzed and summarized to provide answers to the investigative questions found in Chapter I. The investigative questions and the responses obtained follow.

Investigative Questions

1. What type of software support cost estimates (outputs) are provided by the Air Force Cost Center "preferred" software support cost estimating models?

The information provided by the models indicates there are five basic outputs of the models. The four outputs are as follows:

a. Annual Cost of Project - The annual cost for supporting a given software program.

b. Total Cost of Project - The total cost for supporting a given software program.

c. Annual Man-months - The number of man-months used by an organization during the year to support a given software program.

d. Months to Complete - The total number of calendar months required to complete a given software project. This data item can be computed by subtracting the Project Start Date from the Project End Date.

2. What type of software support data (inputs) are required for the Air Force Cost Center "preferred" software support cost estimating models?

The "preferred" models were examined to determine the required support cost inputs. Table 2 shows the data (inputs) which are used, either directly or indirectly, by the "preferred" models. These data items were selected by reviewing the "preferred" models. REVIC was the first model examined, and the following data items were selected:

a. Lines of Code - The number of source instructions in the program, including executable, job control language, format and data declaration statements. Excluded are comments and unmodified utility software.

TABLE 2

MODEL FROM WHICH INPUTS WERE TAKEN

Item Description	REVIC	SASET	SEER	PRICE-S
Lines of Code	X	X	X	X
Analyst's Capability	X		X	X
Pgm Team Capability	X	X	X	X
Project Appl Exp	X	X	X	X
Language Experience	X	X	X	X
Execute Time Const	X	X	X	X
Main Storage Const	X	X	X	X
Virtual Machine Vol	X		X	
Required Software Rel	X	X		X
Data Base Size	X			
Soft Prod Complexity	X	X	X	X
Security Application	X	X		
Mdrn Design Practices	X	X	X	
Use Of Software Tools	X	X	X	X
Software Config Items	X	X		X
Hours Per Manmonth	X	X	X	X
Cost Per Manhour	X	X	X	X
Annual Change Traffic	X			
Commented Document		X		X
Amount of Document		X		X
Number of Oper Copies		X		X
Age of Software		X		
Interfaces		X		
COTS Interfaces		X		
Support Facilities		X	X	
Military Standard		X	X	
Support Locations			X	
Language Used	X	X	X	X
Class of Software	X	X	X	X
Growth of Software			X	X
Project Start Date		X	X	X
Project End Date		X	X	X

b. Analyst's Capability - Measure of the systems engineering capability of the maintenance analysts as a team average. The percentile in which the analysts rate compared to other software maintenance analysts in the field.

c. Programming Team Capability - Measure of the capability of the programmers performing the detailed design and writing/testing the physical code. The percentile in which the programmers rate compared to other programmers in the field.

d. Project Application Experience - Measure of the familiarity of the design and development team with the specific program. The years of experience in this area the average team member has.

e. Language Experience - Measure of the design and programming team's experience with the language. The years of experience the team has with the language.

f. Execution Time Constraints - Measure of the approximate percentage of available execution time that is used by the software.

g. Main Storage Constraints - Measure of the amount of constraint imposed on the software due to main memory limitations in the target computer. The percentage of main memory utilized.

h. Virtual Machine Volatility - Measure of the amount of changes the host and target computers are experiencing during the support phases. Number of changes in a year.

i. Required Software Reliability - Measure of the required reliability of the finished software. What effect would a software failure have on the user? (Slight Inconvenience, Easily Recoverable Loss, Moderate Recoverable Loss, For MIL-STD/High Financial Loss, Loss of Life)

j. Data Base Size - The size of the data base to be maintained and manipulated.

k. Software Product Complexity - Measure of the complexity of the software being supported. Category of software project. (Some Hardware Input/Output and Advanced Data Structures, Real-time Applications and Advanced Math, Extremely Complex Scientific Processing)

l. Classified Security Application - Measure of the security requirements of a software project. (Confidential, Secret, Top secret)

m. Modern Programming Practices - Measure of the use of modern programming practices such as structured design, data flow diagrams, data dictionaries, etc. The extent to which these practices are used. (No use, Beginning Use, Some Use, General Use, Routine Use)

n. Use of Software Tools - Measure of the use of automated software tools such as CASE (computer aided system engineering) tools, ADA Support Environments, etc. (Very few, Basic micro tools, Basic mini tools, Basic maxi tools, Extensive tools with little integration, Moderately integrated tools, Fully integrated tools)

o. Software Configuration Items - Number of software configuration items for the project.

p. Hours Per Man-month - The number of hours per man-man-month used by your organization.

q. Cost Per Man-hour - The average cost for all personnel who work on the software support.

r. Annual Change Traffic - The percentage of existing code that is changed each year.

The second model reviewed was SASET. Because some of the inputs found in the model were the same or similar to REVIC it was determined the data items should be combined to keep the number of data items in the list manageable. The software support inputs found in SASET that were not the same, or were not similar to REVIC are as follows:

a. Commented Documentation - The extent to which the program code is commented. (Very Little, Scattered, Selected, Extensive)

b. Amount of Documentation - The measure of the amount of support documentation available. (Little, Some, Average, Large)

c. Number of Operational Copies - The number of versions being maintained.

d. Age of Software - How long has it been since initial version was released.

e. Interfaces - The number of external interfaces, such as other programs or systems.

f. Commercial Off The Shelf (COTS) Interfaces - The impact, if any, due to integrating with COTS. (Many, Some, Few, None)

g. Support Facilities - The availability of hardware used in the software support effort. (Unavailable due to high utilization, shared hardware, good availability, dedicated hardware)

SEER was the next "preferred" cost model examined. As with the SASET, equivalent and similar inputs were grouped together. The following are the software support inputs that were not grouped with the previously selected data items:

a. Military Standard - The military standard the software is maintained under. (2167, 2167A, 1679, 1703, 7935 etc)

b. Support Locations - Number of locations the software is being maintained.

c. Language Used - Type of programming language used for the project.

d. Class of Software - Type of system for which the software is used. (Manned Flight, Unmanned Flight, Avionics, Ground, Commercial)

e. Growth of Software - The percentage increase in the number of lines of code during the life of the program as a result of software support.

The final model reviewed was PRICE-S. Again, equivalent and similar data items were grouped with inputs

from the other models. The following are the unique PRICE S inputs:

a. Project Start Date - The start date of the subject project.

b. Project End Date - The end date of the subject project.

3. What types of mission critical software support cost data and software support cost-related data are currently collected or can be collected by the Air Force?

The respondents were provided with the list of data items that was developed by examining the "preferred" models, and were asked what data is currently collected. If a data item was not collected, the respondents were asked how difficult it would be to collect. Table 3 provides a summary of the respondent's responses. The majority of data items are not collected, but fell into the "easy to collect" category. The data items that did not fall into that category are discussed below. There were ten data items which the majority of respondents stated they collected. These items are as follows:

a.) Lines of Code

b.) Language Used

c.) Interfaces

d.) Software Configuration Items

e.) Hours per Man-month

f.) Cost per Man-hour

g.) Project Start Date

TABLE 3

SUMMARY OF RESPONSES FOR PART I OF QUESTIONNAIRE

Item Description	Currently Collect	Easy to Collect	Difficult to Collect	Cannot Collect
Lines Of Code	4	3	1	-
Language Used	5	4	-	-
Class of Software	4	5	-	-
Analysts Cpblty	-	4	5	-
Pgm Team Cpblty	-	4	5	-
Project Appl Exp	-	6	3	-
Language Exp	1	5	3	-
Execute Time Const	1	4	3	1
Main Stg Const	1	6	2	-
Virt Machine Vol	2	3	4	-
Req Software Rel	3	5	1	-
Data Base Size	3	5	1	-
Soft Prod Cmplxty	-	7	2	-
Interfaces	4	3	2	-
COTS Interfaces	2	5	2	-
Security Appl	4	4	1	-
Mdrn Dsgn Pract	1	7	1	-
Use Of Soft Tools	1	7	1	-
Military Standard	3	6	-	-
Age Of Software	2	7	-	-
Soft Config Items	4	3	2	-
Support Facilities	1	6	2	-
Support Locations	2	6	1	-
Commented Doc	-	5	4	-
Amount of Doc	-	7	2	-
Num of Op Copies	3	4	2	-
Hours Per Manmonth	5	2	2	-
Cost Per Manhour	7	1	1	-
Annual Chg Traffic	3	2	4	-
Growth of Software	2	3	4	-
Project Start Date	8	-	1	-
Project End Date	8	1		-
Annual Proj Cost	6	-	3	-
Total Proj Cost	6	-	3	-

h.) Project End Date

i.) Annual Cost of Project

j.) Total Cost of Project

The majority of respondents believed it would be difficult for their organization to collect two of the data listed: the analyst's capability and the programming team's capability. Two respondents felt collecting data on the analyst's and programming team's capability was difficult due to a lack of a standard method to measure the capabilities. Two respondents believed it would be difficult because there are a large number of analysts and programmers on a team and in an organization, and the turnover can be extremely high, making personnel tracking very difficult. Another respondent felt the analysts and programmers may feel as though the information will be used against them in some way. All respondents did agree, however, that they could provide an educated guess as to the capability of their analysts and programmers.

Virtual machine volatility was classified as "difficult to collect" by four of the respondents. Two respondents cited the lack of a standard method of measurement; and two respondents felt the data was very erratic and laborious to track, especially for larger programs.

The amount of documentation, annual change traffic and the growth of software were included in the "difficult to collect" category by four respondents. A lack of automated software tools was cited as a major difficulty by three

respondents, and one respondent believed it would be difficult due to the large number of programs managed by the support agency.

The major factors hindering the collection of other data items were: a lack of standard methods for measuring the data items, a lack of automated software tools, and the large number of programs for which the data would need to be collected. Although there were some data items classified as "difficult to collect", it appears that this situation can be overcome. All respondents agreed that if the requirement for actual data was relaxed in these few cases, and an adequate estimate could be substituted, the data could be provided with much less effort, making the data items "easy to collect". In addition, the respondents believed that if the data was requested for a small number of programs, and a clear explanation of what was required was included, the data could be provided without expending a great deal of effort, again moving the classification to "easy to collect". The individual responses of the respondents are contained in Appendix E.

4. What are the current procedures used by Air Force organizations for gathering historical mission critical software support cost data and software support cost-related data?

The procedures for collection of post-deployment software support data varied considerably among respondents. Four respondents stated no procedures were currently in

place, but some data was available from audits in the past and requests from headquarters regarding software support data.

One respondent uses a data base on a personal computer to store data, and sends out forms annually to each of the program teams asking for inputs to the data base. This manual system is scheduled to be automated next year, allowing program managers to access the data base via a local area network (LAN). This will allow the managers to review and make inputs to the data base.

Another respondent obtains PDSS data by accessing a large data base that contains cost per man-hour, time and attendance. In addition, the respondent collects data by examining AFLC Form 75s. (The AFLC Form 75 will be discussed in more detail later.)

Three other respondents have procedures for the collection of software support cost data; but, no procedures are in place for collecting software support cost-related data. Two of these respondents have large data bases which track man-hours spent on a given project. The third respondent collects cost data with the use of a cost data form which is filled out by the software project supporters. The form contains information for the man-hours worked on a specific program. The respondent indicated the form, as it currently exists, does not meet the needs of the user and needs to be revised to provide more accurate data concerning software costs.

The interviews indicate there is no standard procedure for collecting software support cost data at this time. All respondents have an internal system or no system at all to track the data. The AFLC Form 75 does contain some software support cost data and some software support cost-related data, but is used only within AFLC and is only being utilized for tracking purposes by one of the five Air Logistics Centers (ALCs), according to the respondents.

The AFLC Form 75, the Computer Program Configuration Sub-Board Item Record, is to be used whenever anyone in AFLC is considering making changes to software. The Form 75 is prepared as a result of a discrepancy report that has been submitted by the user or the maintainer in response to discovered errors, changing needs, or new requirements. The form is designed to maintain control over the software change proposal system.

This standard form is processed through the Computer Program Configuration Sub-Board (CPCSB). Typical members of this board are division chiefs from the Material Management directorate, Maintenance directorate, and other ALC directorates as they are required (1:13). The form is used for many purposes to include administrative, technical, and managerial requirements.

The form has a variety of entries, including administrative data, background information, a verbal description of the change, an impact statement, and a

section for required schedule and cost for the change (1:59-60). The problem with the system, as it currently exists, is that many times the Form 75 is not updated at the completion of the software change; therefore, the schedules and cost figures are not accurate and are of little help to those individuals attempting to locate software support cost data.

5. In the opinion of knowledgeable personnel in the area, what historical mission critical software cost data and software support cost-related data, should be gathered, and how should these data be gathered?

All respondents felt all data items compiled by examining the models were items that should be collected. The respondents also identified two data items which they believed should be included in the data item list. The items included project name and the software support agency performing the work. Another respondent believed that, as the data collection systems progress, it would be wise to further break out man-hours for support by phases, such as requirements analysis, design, coding, testing and implementation.

The question of how data should be collected can be divided into two concerns: short term and long term. For the short term, all respondents believed the most effective way to collect data was to have software program team leaders complete a form which lists all the required data items. However, the respondents did not agree on the

frequency of the distribution of the form. Two respondents believed the form should be distributed monthly, three believed distributing the form quarterly was best, three believed semi-annually was enough, and one respondent felt annual distribution would meet the needs of the organization. After the data is gathered, it must be stored. Seven respondents believed a data base on a personal computer is the best choice for the short term. Two respondents did not see the need to automate at all, stating that storing the forms in a file cabinet would be adequate.

For long term data collection procedures, seven of the respondents believed that systems should eventually be moved to a large data base that can be accessed via a LAN so system users could review the data, and specified personnel could make inputs to the system. Again, the respondents disagreed on how frequently the data base should be updated. Two respondents believed the data base should be updated as required. One believed the data base should be updated quarterly, and three others believed updates should take place semiannually. The seventh respondent believed updates should take place as required, but the updates should occur annually at a minimum. Two other respondents did not believe there was a need to automate the collection effort, largely due to the small number of programs being managed. They believed automating the system would take more time and money to develop and maintain than it was worth.

Another item of interest relating to data collection procedures that was cited by two respondents was the standardization of the data being collected. They believed standardized data would make it easier to share data among organizations, therefore making the data more useful. When data is collected independently and there is no standard form of collection, it is nearly impossible to summarize. One respondent also mentioned that maintaining the integrity of the data base is crucial. A data base is only as good as the data that is provided. Upper management should stress the importance of providing quality data to the system, and the data base should be monitored very closely for compliance to standards.

V. Recommendations

Preface

The content of this chapter is based on the findings and conclusions from the previous chapter. The recommendations listed below are followed by a list of possible areas of further study. The areas that are identified are related to the topic.

Suggestions and Recommendations

The organized collection of software support cost data and software support cost-related data should begin as soon as possible. These data are collectable by the support organization, but currently there is no standard data set, no well-defined measure for the items in the data set, and no standard procedures in place for the collection of data. The effort should start with the definition of a standardized data set since a standardized data set will allow the data to be useful to a larger group of people, whether they are members of other software support teams in the same agency, personnel at another software support agency, or cost personnel. In addition, summarizing and reporting this information will be much less painstaking. There may be a few items of specific interest to an organization that can be collected in addition to the

standard data set, but there needs to be a minimum acceptable set of data that is tracked.

A situation to avoid is attempting to collect too many data items. When individuals are asked to provide too much information the data may be biased. In addition, the data set can become very cumbersome to work with, causing the system to be unmanageable. The data set presented in this study provides a starting point along with the additional two items cited by the respondents in Chapter 4, which are the project name, and the name of the agency supporting the software. A proposed data collection form is included in Appendix F.

The next step is to ensure that each item in the data set is well defined; that is, there is a standard measure for the data being collected. For example questions such as, what will be the standard definition for lines of code? Do lines of code include comments or not? Is only executable code counted? These questions must be answered early in the process. The explanations provided by the various models of what is to be measured for given inputs is a good place to start. Once it is determined how each item should be measured, the measures should be documented and provided to the individuals who are actually providing inputs to the data base.

The collection effort should start with a small number of selected projects for which the largest amount of quality data is available. This allows the users to determine where

difficulties are going to be encountered and explore possible solutions to these problems.

The collection effort should begin by distributing forms listing the required data items. The forms should be disseminated to project team leaders by the data collection focal point so the data set may be reviewed.

Following distribution, an appointment should be arranged between the focal point and project team leader to review and complete the form. This allows the focal point the opportunity to clarify any questions or concerns the team leader may have. The focal point can also get a feel for the difficulties in collecting information for the data set. Additionally, if the focal point is present when the form is completed, it is likely the responses will be more accurate.

At times the team leader may be required to provide an estimate or educated guess for a data item. As some of the respondents reported, there may be some data items that are very difficult to provide exact measures for. In these cases, the fact that the input is an estimate should be documented. This information could be very useful in the future when models are being calibrated.

Following data collection, the focal point must store the data for future use. In many cases, the best place to store this data is in a data base on a personal computer accessible to the focal point. In some cases the number of

programs being tracked is not large enough to justify automation.

The data base should be updated semi-annually, to the largest extent possible. This may not be reasonable or acceptable for some organizations, but is a good guide. If a data collection is not performed frequently enough, vital data may not be included when needed.

As the system matures and the project team leaders become more comfortable with the data set and what data are required, the focal point will not need to be present when the team leader completes the form.

As the collection effort grows and more programs are included, plans should be made to automate the procedures, removing some of the workload from the data collection focal point. This can be done by moving the small data base on the personal computer to a larger, user-friendly system which is connected to a network with many users. In the case of AFLC, for example, current systems which connect the ALCs should be examined. This allows team leaders the ability to review the data base and provide the required data via their computers, again easing the work load. Moving to a larger system with network capability also allows a larger number of users access to the data, whether it be another project team, another support agency, or interested cost personnel. Obviously, a careful review of who has access to the system and who can make inputs to the system should be undertaken.

The integrity of the data base is vital. It does not do any good to have a system for data collection, if the data contained in the data base is incomplete or not accurate. Great care should be taken in monitoring and maintaining the data base. As for the frequency of update, again, semi-annual, mandatory updates should be provided and team leaders should be encouraged to update data items as they change.

The major factors in making this type of system work are communications and a commitment from upper management. If personnel at the working level perceive strong management commitment, understand why data are being collected and believe this collection effort will benefit them, they will work very hard to make this system succeed. Indeed, a similar type of system is working, as cited by one of the respondents.

Further Study

From the research involved in this study, other areas of interest associated with the topic were discovered. Due to the scope of the research project, this researcher was unable to investigate the areas provided below.

1. Perform an actual data collection using the proposed form and the procedures defined in this study, to determine the adequacy of the data set and proposed procedures.

2. Examine and identify possible ways to share this and other software-related data between the various software organizations in DoD.

3. AFLC is in the process of developing a software support cost model (11). With the addition of the unique inputs and outputs for the AFLC model, perform a data collection and calibration effort for the model.

Appendix A: List of Respondents

The following is a general grouping of respondents who participated in the study. All individuals which participated are knowledgeable in the area of post-deployment software support on MCCR software. The names of the individuals who participated were withheld to preserve the anonymity promised.

1. Air Force Logistics Command had five respondents representing the following agencies:

   A. Oklahoma City Air Logistics Center

   B. Ogden Air Logistics Center

   C. Warner Robbins Air Logistics Center

   D. Sacramento Air Logistics Center

   E. Newark Air Force Station

2. The Military Airlift Command had one respondent from Scott Air Force Base.

3. The Space Command had one respondent from Falcon Air Force Station.

4. The Strategic Air Command had one respondent from Offut Air Force Base.

5. The Tactical Air Command had one respondent from Langley Air Force Base.

Appendix B: Interview Confirmation Letter

The attached telephone confirmation letter was sent to each respondent to provide available dates for interviews. The letter provided pertinent information in the case it was necessary for respondent to contact the researcher.

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY
WRIGHT-PATTERSON AIR FORCE BASE OH 45433-6583

REPLY TO ATTN OF: AFIT/LSG (Capt Brent Barber/GSS-91D)

SUBJECT: Thesis Telephone Interview

TO: Respondent

Thank you for agreeing to participate in my study of post-deployment software support data collection within the Air Force. I am trying to determine what software support cost data is currently being collected, what data can be collected, and how this data can be collected. As explained in our telephone conversation, the information will be used for my masters thesis in software management at the Air Force Institute of Technology.

I will be telephoning you in the near future to set up a date and time for the interview session. If possible, I will perform all interviews between 6 June 91 and 14 June 91. Please review the enclosed questionnaire before the interview.

Your time and cooperation are greatly appreciated. Your participation will provide needed information to determine the current status and future plans for software support data collection. A copy of your responses and of the thesis will be sent to you upon completion of the study.

If you have any questions or concerns feel free to contact me at Autovon 785-8989, or from 6 June to 14 June at Autovon 785-7870.

Sincerely,

BRENT L. BARBER, Capt, USAF
AFIT Graduate Student

Appendix C: Telephone Interview Form

Questionnaire on Post-Deployment Software Support Cost Data

Questions in Part I of this questionnaire should be answered for each of the data items listed in attachment 1. When this has been completed, proceed to Part II. Part II consists of general questions which apply to the data set listed in attachment 1, as a whole.

Part I

1. Does your organization currently collect the subject data item for all software projects?

-- If you answered yes to question 1 proceed to the next data item listed in attachment 1, otherwise proceed to question 2.

2. How difficult would it be for your organization to collect the subject data item? (Choose from below)

    a.) Would be easy to collect, but is not currently collected

    b.) Would be difficult to collect at this time

    c.) Cannot be collected

-- If you selected "a" please proceed to the next data item listed in attachment 1, otherwise proceed to question 3.

3. What factors hinder or prevent collection of the subject data item?

4. How could the factors which hinder or prevent collection be overcome?

-- Proceed to next data item listed in attachment 1.

Part II

1. What procedures does your organization have in place to enable you to collect software support cost data?

2. Given the purpose of this data set is to help analyze and calibrate various software support cost estimating models;

   i. Is there any data that is not listed in attachment 1, which you believe would be beneficial in collecting?

   ii. In your opinion, what is the best way to collect this software support cost data?

Attachment 1

LIST OF DATA ITEMS

A.) LINES OF CODE - The number of source instructions in the program, including executable, job control language, format and data declaration. Excludes comments and unmodified utility software.

B.) LANGUAGE USED - Type of programming language used for the project.

C.) CLASS OF SOFTWARE - Type of system the software is used on. (Manned Flight, Unmanned Flight, Avionics, Ground, Commercial)

D.) ANALYSTS CAPABILITY - Measure of the systems engineering capability of the maintenance analysts as a team average. The percentile in which the analysts rate compared to other software maintenance analysts in the field.

E.) PROGRAMMING TEAM CAPABILITY - Measure of the capability of the programmers performing the detailed design and writing/testing the physical code. The percentile in which the programmers rate compared to other programmers in the field.

F.) PROJECT APPLICATION EXPERIENCE - Measure of the familiarity of the design and development team with the specific program. The years of experience in this area the average team member has.

G.) LANGUAGE EXPERIENCE - Measure of the design and programming team's experience with the language. The years of experience the team has with the language.

H.) EXECUTION TIME CONSTRAINTS - Measure of the approximate percentage of available execution time that is used by the software.

I.) MAIN STORAGE CONSTRAINTS - Measure of the amount of constraint imposed on the software due to main memory limitations in the target computer. The percentage of main memory utilized.

J.) VIRTUAL MACHINE VOLATILITY - Measure of the amount of changes the host and target computers are experiencing during the support phases. Number of changes in a year.

K.) REQUIRED SOFTWARE RELIABILITY - Measure of the required reliability of the finished software. What effect would a software failure have on the user. (Slight Inconvenience, Easily Recoverable Loss, Moderate Recoverable Loss, For MIL-STD/High

Financial Loss, Loss of Life)

L.) DATA BASE SIZE - The size of the data base to be maintained and manipulated. DB BYTES/PROGRAM DSI

M.) SOFTWARE PRODUCT COMPLEXITY - Measure of the complexity of the software being supported. Category of software project. (Some Hardware Input/Output and Advanced Data Structures, Real-time Applications and Advanced Math, Extremely Complex Scientific Processing)

N.) INTERFACES - The number of external interfaces, such as other programs or systems.

O.) COMMERCIAL OFF THE SHELF (COTS) INTERFACE - The impact, if any, due to integrating with COTS. (Many, Some, Few, None)

P.) CLASSIFIED SECURITY APPLICATION - Measure of the security requirements of a software project. (Confidential, Secret, Top secret)

Q.) MODERN PROGRAMMING PRACTICES - Measure of the use of modern programming practices such as structured design, data flow diagrams, data dictionaries, etc. The extent to which these practices are used. (No use, Beginning Use, Some Use, General Use, Routine Use)

R.) USE OF SOFTWARE TOOLS - Measure of the use of automated software tools such as CASE (computer aided system engineering) tools, ADA Support Environments, etc. (Very few, Basic micro tools, Basic mini tools, Basic maxi tools, Extensive tools with Little integration, Moderately integrated tools, Fully integrated tools)

S.) MILITARY STANDARD - The military standard the software is maintained under. (2167, 2167A, 1679, 1703, 7935 etc)

T.) AGE OF SOFTWARE - How long has it been since initial version was released.

U.) SOFTWARE CONFIGURATION ITEMS - Number of software configuration items for the project.

V.) SUPPORT FACILITIES - The availability of hardware used in the software support effort. (Unavailable due to high utilization, shared hardware, availability good, dedicated hardware)

W.) SUPPORT LOCATIONS - Number of locations the software is

being maintained.

X.) COMMENTED DOCUMENTATION - The extent to which the program code is commented. (Very Little, Scattered, Selected, Extensive)

Y.) AMOUNT OF DOCUMENTATION - The measure of the amount of support documentation available. (Little, Some, Average, Large)

Z.) NUMBER OF OPERATIONAL COPIES - The number of versions being maintained.

AA.) HOURS PER MAN-MONTH - The number of hours per man-month used by your organization.

BB.) COST PER MAN-HOUR - The average cost for all personnel who work on the software support.

CC.) ANNUAL CHANGE TRAFFIC - The percentage of existing code that is changed each year.

DD.) GROWTH OF SOFTWARE - The percentage of increase in the number of lines of code as a result of software support.

EE.) PROJECT START DATE - The start date of the subject project.

FF.) PROJECT END DATE - The end date of the subject project.

GG.) ANNUAL COST OF PROJECT - The annual cost for support on the given program.

HH.) TOTAL COST OF PROJECT - Total support cost for a finished project or block change.

Appendix D: Letter of Appreciation

The following letter was sent to each respondent who participated in the study.

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY
WRIGHT-PATTERSON AIR FORCE BASE OH 45433-6583

REPLY TO ATTN OF: AFIT/LSG (Capt Brent Barber/GSS-91D)

SUBJECT: Thesis Telephone Interview

TO: Respondent

I wish to extend my appreciation for your participation in my study of software support data collection within the Air Force. Your responses in the telephone interview will remain anonymous. The information which you have provided will contribute to the software support data collection efforts of the Air Force.

Thank you again for your time and cooperation with this study. A copy of the completed study will be available through the Defense Technical Information Center (DTIC). The title of the study is:

INVESTIGATIVE SEARCH OF QUALITY HISTORICAL SOFTWARE SUPPORT COST DATA AND SOFTWARE SUPPORT COST-RELATED DATA.

Sincerely,

BRENT L. BARBER, Capt, USAF
AFIT Graduate Student

Appendix E: Individual Responses Part I of Questionnaire

The following are the individual responses which were provided for Part I of the questionnaire.

RESPONSES FOR PART I OF QUESTIONNAIRE
(OC-ALC/AFLC RESPONDENT)

Item Description	Currently Collect	Easy to Collect	Difficult to Collect	Cannot Collect
Lines Of Code		X		
Language Used		X		
Class of Software		X		
Analysts Cpblty			X	
Pgm Team Cpblty			X	
Project Appl Exp		X		
Language Exp		X		
Execute Time Const			X	
Main Stg Const		X		
Virt Machine Vol			X	
Req Software Rel		X		
Data Base Size		X		
Soft Prod Cmplxty			X	
Interfaces			X	
COTS Interfaces			X	
Security Appl		X		
Mdrn Dsgn Pract		X		
Use Of Soft Tools		X		
Military Standard		X		
Age Of Software		X		
Soft Config Items		X		
Support Facilities			X	
Support Locations		X		
Commented Doc			X	
Amount of Doc			X	
Num of Op Copies		X		
Hours Per Manmonth	X			
Cost Per Manhour	X			
Annual Chg Traffic			X	
Growth of Software			X	
Project Start Date			X	
Project End Date		X		
Annual Proj Cost			X	
Total Proj Cost			X	

RESPONSES FOR PART I OF QUESTIONNAIRE
(OO-ALC/AFLC RESPONDENT)

Item Description	Currently Collect	Easy to Collect	Difficult to Collect	Cannot Collect
Lines Of Code	X			
Language Used	X			
Class of Software	X			
Analysts Cpblty		X		
Pgm Team Cpblty		X		
Project Appl Exp		X		
Language Exp		X		
Execute Time Const		X		
Main Stg Const		X		
Virt Machine Vol		X		
Req Software Rel		X		
Data Base Size		X		
Soft Prod Cmplxty		X		
Interfaces	X			
COTS Interfaces		X		
Security Appl	X			
Mdrn Dsgn Pract	X			
Use Of Soft Tools	X			
Military Standard		X		
Age Of Software	X			
Soft Config Items	X			
Support Facilities		X		
Support Locations		X		
Commented Doc		X		
Amount of Doc		X		
Num of Op Copies	X			
Hours Per Manmonth	X			
Cost Per Manhour	X			
Annual Chg Traffic		X		
Growth of Software		X		
Project Start Date	X			
Project End Date	X			
Annual Proj Cost	X			
Total Proj Cost	X			

RESPONSES FOR PART I OF QUESTIONNAIRE
(WR-ALC/AFLC RESPONDENT)

Item Description	Currently Collect	Easy to Collect	Difficult to Collect	Cannot Collect
Lines Of Code			X	
Language Used	X			
Class of Software	X			
Analysts Cpblty			X	
Pgm Team Cpblty			X	
Project Appl Exp			X	
Language Exp			X	
Execute Time Const			X	
Main Stg Const			X	
Virt Machine Vol			X	
Req Software Rel			X	
Data Base Size		X		
Soft Prod Cmplxty		X		
Interfaces		X		
COTS Interfaces		X		
Security Appl		X		
Mdrn Dsgn Pract		X		
Use Of Soft Tools		X		
Military Standard		X		
Age Of Software	X			
Soft Config Items	X			
Support Facilities			X	
Support Locations		X		
Commented Doc			X	
Amount of Doc		X		
Num of Op Copies	X			
Hours Per Manmonth	X			
Cost Per Manhour	X			
Annual Chg Traffic			X	
Growth of Software			X	
Project Start Date	X			
Project End Date	X			
Annual Proj Cost	X			
Total Proj Cost	X			

RESPONSES FOR PART I OF QUESTIONNAIRE
(SAC RESPONDENT)

Item Description	Currently Collect	Easy to Collect	Difficult to Collect	Cannot Collect
Lines Of Code	X			
Language Used	X			
Class of Software	X			
Analysts Cpblty			X	
Pgm Team Cpblty			X	
Project Appl Exp			X	
Language Exp			X	
Execute Time Const				X
Main Stg Const		X		
Virt Machine Vol			X	
Req Software Rel	X			
Data Base Size	X			
Soft Prod Cmplxty		X		
Interfaces	X			
COTS Interfaces	X			
Security Appl	X			
Mdrn Dsgn Pract		X		
Use Of Soft Tools		X		
Military Standard	X			
Age Of Software		X		
Soft Config Items	X			
Support Facilities		X		
Support Locations		X		
Commented Doc		X		
Amount of Doc		X		
Num of Op Copies	X			
Hours Per Manmonth	X			
Cost Per Manhour	X			
Annual Chg Traffic	X			
Growth of Software	X			
Project Start Date	X			
Project End Date	X			
Annual Proj Cost	X			
Total Proj Cost	X			

RESPONSES FOR PART I OF QUESTIONNAIRE
(NEWARK AFS RESPONDENT)

Item Description	Currently Collect	Easy to Collect	Difficult to Collect	Cannot Collect
Lines Of Code		X		
Language Used		X		
Class of Software		X		
Analysts Cpblty		X		
Pgm Team Cpblty		X		
Project Appl Exp		X		
Language Exp		X		
Execute Time Const		X		
Main Stg Const		X		
Virt Machine Vol	X			
Req Software Rel		X		
Data Base Size		X		
Soft Prod Cmplxty		X		
Interfaces		X		
COTS Interfaces		X		
Security Appl		X		
Mdrn Dsgn Pract		X		
Use Of Soft Tools		X		
Military Standard		X		
Age Of Software		X		
Soft Config Items		X		
Support Facilities		X		
Support Locations		X		
Commented Doc		X		
Amount of Doc		X		
Num of Op Copies		X		
Hours Per Manmonth		X		
Cost Per Manhour	X			
Annual Chg Traffic		X		
Growth of Software		X		
Project Start Date	X			
Project End Date	X			
Annual Proj Cost	X			
Total Proj Cost	X			

RESPONSES FOR PART I OF QUESTIONNAIRE
(MAC RESPONDENT)

Item Description	Currently Collect	Easy to Collect	Difficult to Collect	Cannot Collect
Lines Of Code			X	
Language Used		X		
Class of Software		X		
Analysts Cpblty			X	
Pgm Team Cpblty			X	
Project Appl Exp			X	
Language Exp		X		
Execute Time Const		X		
Main Stg Const		X		
Virt Machine Vol		X		
Req Software Rel			X	
Data Base Size		X		
Soft Prod Cmplxty			X	
Interfaces	X			
COTS Interfaces		X		
Security Appl	X			
Mdrn Dsgn Pract		X		
Use Of Soft Tools		X		
Military Standard	X			
Age Of Software		X		
Soft Config Items			X	
Support Facilities			X	
Support Locations	X			
Commented Doc			X	
Amount of Doc			X	
Num of Op Copies			X	
Hours Per Manmonth	X			
Cost Per Manhour	X			
Annual Chg Traffic			X	
Growth of Software			X	
Project Start Date	X			
Project End Date	X			
Annual Proj Cost	X			
Total Proj Cost	X			

RESPONSES FOR PART I OF QUESTIONNAIRE
(SPACE COMMAND RESPONDENT)

Item Description	Currently Collect	Easy to Collect	Difficult to Collect	Cannot Collect
Lines Of Code	X			
Language Used	X			
Class of Software	X			
Analysts Cpblty		X		
Pgm Team Cpblty		X		
Project Appl Exp		X		
Language Exp		X		
Execute Time Const		X		
Main Stg Const	X			
Virt Machine Vol	X			
Req Software Rel	X			
Data Base Size	X			
Soft Prod Cmplxty		X		
Interfaces	X			
COTS Interfaces	X			
Security Appl		X		
Mdrn Dsgn Pract		X		
Use Of Soft Tools		X		
Military Standard		X		
Age Of Software		X		
Soft Config Items	X			
Support Facilities	X			
Support Locations	X			
Commented Doc		X		
Amount of Doc		X		
Num of Op Copies	X			
Hours Per Manmonth		X		
Cost Per Manhour		X		
Annual Chg Traffic	X			
Growth of Software	X			
Project Start Date	X			
Project End Date	X			
Annual Proj Cost			X	
Total Proj Cost			X	

RESPONSES FOR PART I OF QUESTIONNAIRE
(SAC RESPONDENT)

Item Description	Currently Collect	Easy to Collect	Difficult to Collect	Cannot Collect
Lines Of Code	X			
Language Used	X			
Class of Software	X			
Analysts Cpblty			X	
Pgm Team Cpblty			X	
Project Appl Exp			X	
Language Exp			X	
Execute Time Const				X
Main Stg Const		X		
Virt Machine Vol			X	
Req Software Rel	X			
Data Base Size	X			
Soft Prod Cmplxty		X		
Interfaces	X			
COTS Interfaces	X			
Security Appl	X			
Mdrn Dsgn Pract		X		
Use Of Soft Tools		X		
Military Standard	X			
Age Of Software		X		
Soft Config Items	X			
Support Facilities		X		
Support Locations		X		
Commented Doc		X		
Amount of Doc		X		
Num of Op Copies	X			
Hours Per Manmonth	X			
Cost Per Manhour	X			
Annual Chg Traffic	X			
Growth of Software	X			
Project Start Date	X			
Project End Date	X			
Annual Proj Cost	X			
Total Proj Cost	X			

RESPONSES FOR PART I OF QUESTIONNAIRE
(TAC RESPONDENT)

Item Description	Currently Collect	Easy to Collect	Difficult to Collect	Cannot Collect
Lines Of Code	X			
Language Used	X			
Class of Software		X		
Analysts Cpblty		X		
Pgm Team Cpblty		X		
Project Appl Exp		X		
Language Exp	X			
Execute Time Const	X			
Main Stg Const		X		
Virt Machine Vol		X		
Req Software Rel	X			
Data Base Size	X			
Soft Prod Cmplxty		X		
Interfaces		X		
COTS Interfaces		X		
Security Appl	X			
Mdrn Dsgn Pract		X		
Use Of Soft Tools		X		
Military Standard	X			
Age Of Software		X		
Soft Config Items		X		
Support Facilities		X		
Support Locations		X		
Commented Doc		X		
Amount of Doc		X		
Num of Op Copies		X		
Hours Per Manmonth			X	
Cost Per Manhour			X	
Annual Chg Traffic		X		
Growth of Software		X		
Project Start Date	X			
Project End Date	X			
Annual Proj Cost			X	
Total Proj Cost			X	

Appendix F: Proposed Software Support Data Collection Form

1. Project Name _____

   a. Single CSCI [] c. Other []
   b. Multiple CSCI []

2. Software Support Organization _____

3. Source Lines of Code (If Single CSCI - Complete CSCI #1)

   CSCI #1 _____
   CSCI #2 _____
   CSCI #3 _____
   CSCI #4 _____
   CSCI #5 _____

4. Data Base Size(Bytes) _____

5. Programming Language(s)

   a. _____ %____
   b. _____ %____
   c. _____ %____

6. Class of Software

   a. Military Ground []
   b. Military Mobile (ship/van) []
   c. Commercial Ground []
   d. Commercial Avionics []
   e. Mil-Spec Avionics []
   f. Missile []
   g. Unmanned Space []
   h. Manned Space []

7. Project Start Date (MM/YY) _____

8. Project End Date (MM/YY) _____

	VL	LO	NM	HI	VH	XH
9. Analysts Capability	[]	[]	[]	[]	[]	
10. Programming Team Capability	[]	[]	[]	[]	[]	
11. Project Application Exp	[]	[]	[]	[]	[]	
12. Language Experience	[]	[]	[]	[]	[]	
13. Execution Time Constraints		[]	[]	[]	[]	[]

	VL	LO	NM	HI	VH	XH
14. Main Storage Constraints			[]	[]	[]	[]
15. Virtual Machine Volatility	[]	[]	[]	[]	[]	[]
16. Software Product Complexity	[]	[]	[]	[]	[]	[]
17. Required Software Reliability	[]	[]	[]	[]	[]	

18. Number of Internal Interfaces _____

19. Impacts of Integrating with Commercial Of the Shelf Software

a. Many []
b. Some []
c. Few []
d. None []

20. Classified Security Application

a. Unclassified []
b. Confidential []
c. Secret []
d. Top Secret []

21. Modern Programming Practices

a. No Use []
b. Use By a Few Top People []
c. Use By Some People []
d. Use By Most People []
e. Use By All, With Training Emphasized []

22. Use of Software Tools

a. Very Few []
b. Basic Micro Tools []
c. Basic Mini Tools []
d. Basic Maxi Tools []
e. Extensive Tools/Little Integration []
f. Moderately Integrated Tools []
g. Fully Integrated Tools []

23. Military Standard Used In Development

a. Commercial []
b. DOD-STD-2167(Tailored) []
c. DOD-STD-2167A(Tailored) []
d. DOD-STD-2167(Full) []
e. DOD-STD-2167A(Full) []
f. DOD-STD-1703 []
g. DOD-STD-1679 []
h. Other _____

24. Age of Software (Years Since First Release) _____

25. Number of Computer Software Configuration Items _____

26. Support Facilities

a. Hardware Unavailable Due to High Utilization []
b. Shared Hardware []
c. Good Availability of Hardware []
d. Dedicated Hardware []

27. Number of Support Locations _____

28. Commented Code

a. Very Little []
b. Scattered []
c. Selected []
d. Extensive []

29. Amount of Support Documentation

a. Little []
b. Some []
c. Average []
d. Large []

30. Number of Operational Copies _____

31. Cost Per Man-hour Used by Supporting Organization _____

32. Annual Change Traffic %_____

33. Growth of Software %_____

34. Hours Per Man-month for Project _____

35. Annual Cost of Project _____

36. Total Cost of Project _____

Description of Data Items

1. Program Name - Enter the name of the program.

2. Software Support Organization - Enter the name of the organization(s) supporting the software.

3. Source Lines of Code - The number of source instructions in the program for each CSCI, including executable, job control language, format and data declaration. Excludes comments and unmodified utility software.

4. Data Base Size - Enter the total data base size in bytes.

5. Programming Language Used - Enter the programming language(s) used for the project. If more than one programming language is used, indicate the percentage of the total for each language selected.

6. Class Of Software - Select the operating environment from the list which best describes the primary mission of the software being supported.

7. Project Start Date - Enter the start date of the subject project.

8. Project End Date - Enter the end date of the subject project.

9. Analysts Capability - The measure of the systems engineering capability of the maintenance analysts as a team average. Mark the two letter code which corresponds to the capability of the analysts compared to other software maintenance analysts in the field.

Rating	Example
VL	New personnel with no experience
LO	Functional team with low effectivity
NM	Average team with nominal effectivity
HI	Strong team with good effectivity
VH	Strong team with many top people

10. Programming Team Capability - The measure of the capability of the programmers performing the detailed design and writing/testing the physical code. Mark the two letter code which corresponds to the capabilities of the programmers compared to other programmers in the field.

Rating	Example
VL	New personnel with no experience
LO	Functional team with low effectivity
NM	Average team with nominal effectivity
HI	Strong team with good effectivity
VH	Strong team with many top people

11. Project Application Experience - Measure of the familiarity of the design and development team with the specific program. Mark the two letter code which corresponds to the years of experience in this area the average team member has.

Rating	Example
VL	No Experience (less than 4 months)
LO	Limited Experience (1 year)
NM	Nominal Experience (3 years)
HI	Better Than Average (6 years)
VH	Experts (more than 12 years)

12. Language Experience - Measure of the design and programming team's experience with the language. Mark the two letter code which corresponds to the years of experience the team has with the language.

Rating	Example
VL	Never Used Before
LO	Less than 1 Year Experience
NM	At least 1 Year Experience
HI	2 Years Experience
VH	More Than 2 Years Experience

13. Execution Time Constraints - Measure of the approximate percentage of available execution time that is used by the software. Mark the two letter code which corresponds to the approximate amount of utilization.

Rating	Example
LO	No Constraints on Execution Time
NM	60% Utilization
HI	70% Utilization
VH	85% Utilization
XH	95% Or More Utilization

14. Main Storage Constraints - Measure of the amount of constraint imposed on the software due to main memory limitations in the target computer. Mark the two letter code which corresponds to the percentage of main memory utilized.

Rating	Example
NM	No Memory Constraints
HI	70% Utilization
VH	85% Utilization
XH	95% Or Higher Utilization

15. Virtual Machine Volatility - Measure of the amount of changes the host and target computers are experiencing during the support phases. Mark the two letter code which best describes the frequency of changes.

Rating	Example
VL	No Changes
LO	One Change Per 6 Months
NM	One Change Per 3 Months
HI	One Change Per Month
VH	Several Changes Per Month
XH	Constantly Changing

16. Software Product Complexity - Measure of the complexity of the software being supported. Mark the two letter code which best describes the category of the software project.

Rating	Example
VL	Offline Simple Print Routines
LO	Offline Data Processing
NM	Data Processing and Math Routines
HI	Some Hardware I/O and Math Routines
VH	Real Time Applications and Advanced Math
XH	Extremely Complex Scientific Processing

17. Required Software Reliability - Measure of the required reliability of the finished software. Mark the two letter code that best describes the effect a software failure would have on the user.

Rating	Example
VL	Slight Inconvenience
LO	Easily Recoverable Loss
NM	Moderate Recoverable Loss
HI	For MIL-STD or High Financial Loss
VH	Possible Loss of Life

18. Interfaces - Enter the number of external interfaces, such as other programs or systems.

19. Commercial Off The Shelf (COTS) Interface - Select from the list, the severity of the impacts, if any, due to integrating with COTS.

20. Classified Security Application - Select from the list the security requirements for the software project.

21. Modern Programming Practices - Measure of the use of modern programming practices such as structured design, data flow diagrams, data dictionaries, etc. Select from the list, the extent to which these practices are used.

22. Use of Software Tools - Select from the list the usage of automated software tools such as CASE (computer aided system engineering) tools, ADA Support Environments, etc.

23. Military Standard Used in Development - Select from the list the military standard the software was developed under.

24. Age of Software - Enter the number of years since the initial version was released.

25. Computer Software Configuration Items - Enter the number of computer software configuration items for the project.

26. Support Facilities - Select from the list the choice which best describes the availability of hardware used in the software support effort.

27. Support Locations - Enter the number of locations the software is being maintained.

28. Commented Code - Select from the list the choice which best describes the extent to which the program code is commented.

29. Amount of Support Documentation - Select from the list the choice which best describes the amount of support documentation available.

30. Number of Operational Copies - Enter the number of versions being maintained.

31. Cost Per Man-hour Used by Supporting Organization - Enter the average cost for all personnel who work on the software support.

32. Annual Change Traffic - Enter the percentage of existing code that is changed each year.

33. Growth of Software - Enter the percentage of increase per year in the number of lines of code as a result of software support.

34. Hours Per Man-month for Project - Enter the number of hours per man-month used by your organization for a given project.

35. Annual Cost Of Project - Enter the annual cost for support on the given program.

36. Total Cost of Project - Enter the total support cost for a finished project or block change.

Bibliography

1. Air Force Logistics Command. Management and Support Procedures for Computer Resources Used in Defense Systems. AFLCR 800-21. Wright-Patterson AFB OH, 21 January 1983

2. Boehm, Barry. Software Engineering Economics. Englewood Cliffs NJ: Prentice-Hall, Inc., 1981.

3. Brooks, Frederick P. "No Silver Bullet," IEEE Computer: 10-18 (April 1987).

4. Clark, Gregory A. Software Cost Estimation Models -- Which One to Use?, ACSC Student Report # 86-0545, Maxwell AFB AL, Air Command and Staff College: 1986.

5. Dekker, G.J. A Software-Cost Database for the Development of Aerospace Software, National Aerospace Laboratory Report # MP 82044 U, Amsterdam Netherlands, National Aerospace Laboratory: 1982.

6. -----. Functional Requirements For A Software Cost Database, National Aerospace Laboratory Report # TR 81017 U, Amsterdam Netherlands, National Aerospace Laboratory: 1981.

7. Department of Defense. Defense System Software Development. DOD-STD-2167A. Washington: Government Printing Office, 19 February 1988.

8. -----. MCCR Software Support. MIL-HDBK-347. Washington: Government Printing Office, 22 May 1990.

9. -----. Software Master Plan. Defense Acquisition Board (DAB) Science and Technology Committee Software Working Group, Washington DC: 1990.

10. Department of the Air Force. Life Cycle Management of Computer Resources in Systems. AFR 800-14. Washington DC: HQ USAF, 29 September 1986.

11. Donald, Brad, Air Force Cost Analyst. Telephone interview. Air Force Cost Center, Crystal City VA, 9 November 1990.

12. Emory, William C. Business Research Methods (Third Edition). Homewood IL: Irwin, 1985.

13. Ferens, Daniel V. Class handout distributed in IMGT 676, Software Cost Estimation. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, April 1991.

14. -----. Defense System Software Project Management. Dayton OH: Government Printing Office, 1990.

15. -----. Mission Critical Computer Software Support Management. Dayton OH: Government Printing Office, 1987.

16. -----. "Software Support Cost Estimation: A Key to Budgetary Control." DoD Cost Symposium. Dayton: Department of Defense, 1990.

17. Galorath Associates, SEER Quick Reference Guide, Marina del Rey CA, Galorath Associates: 2 March 1990.

18. Glass, Robert L. and Ronald A. Noiseux. Software Maintenance Guide Book. Englewood Cliffs NJ: Prentice-Hall Inc., 1981.

19. Kile, Raymond L. REVIC Software Cost Estimating Model User's Manual (Version 9.0), Littleton CO, Raymond Kile: 1991.

20. Martin, James and Carma McClure. Software Maintenance: The Problem and Its Solution. Englewood Cliffs NJ: Prentice-Hall Inc., 1983.

21. McBride, Dorothy J. "AFIT Tackles the Software Problem," Air Force Magazine, 80-82 (December 1989).

22. National Bureau of Standards. Software Maintenance Management. Special Publication 500-129. Washington: Government Printing Office, October 1985.

23. Silver, Aaron, et al, SASET User's Guide (Version 2.0), Denver CO, Martin Marietta: February 1990.

24. Sommerville, Ian. Software Engineering (Third Edition). New York: Addison-Wesley Publishing Company, 1988.

25. SYSCON, Avionics Software Support Cost Model: Final Report, September 1980 - November 1982. Contract F33615-80-C-1157. Washington: SYSCON Corporation, 1 February 1983 (AD-A128523).

VITA

Brent L. Barber was born on 29 July 1961 in McPherson, Kansas. He graduated from Northglenn High School in Northglenn, Colorado, in 1979. He attended the University of Northern Colorado, from which he received a Bachelor of Science degree with a major in Business Management (specialty: Management Information Systems) in June 1986. After graduation, he received a regular commission in the USAF and was stationed at Wright-Patterson AFB, Ohio. He began his service as a Configuration Manager for the Propulsion System Program Office (SPO) where he was responsible for configuration control of the F110-100, F100-101 and F110-129 jet engines. In 1988 he became the Development Program Manager for the F110-129 engine program, where he was responsible for all the aspects of the development process, such as budgeting, contracting, engineering, logistics, and safety. He served in this position until 1989 when he took on additional responsibility by taking on the position of Production Program Manager while continuing to serve as the Development Program Manager for the F110-129 engine. He held these positions until May 1990 when he entered the School of Systems and Logistics, Air Force Institute of Technology.

Permanent Address: 3347 B. W. 114th Circle
Westminster, Colorado 80030

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	Dec 91	Master's Thesis

4. TITLE AND SUBTITLE

INVESTIGATIVE SEARCH OF QUALITY HISTORICAL SOFTWARE SUPPORT COST DATA AND SOFTWARE SUPPORT COST-RELATED DATA

5. FUNDING NUMBERS

6. AUTHOR(S)

Brent L. Barber, Capt, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology, WPAFB, OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GSS/LSY/91D-1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution limited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The purpose of the study was to determine what software support cost data and software support cost-related data should and can be collected by the Air Force. In addition, the study was to determine how the subject data should be collected. The collection of this data is necessary in order to measure the accuracy and calibration requirements of the software support cost models used by the Air Force.

The study found there was no standard set of data items, and no standard procedures in place for the collection of this data. In most cases data were available or easily accessible, but the data had not been requested, therefore, it was not collected. A few organizations were collecting data, but mainly for the purpose of reporting to their superiors.

Data were obtained by reviewing the Air Force "preferred" software support cost models and through semi-structured telephone interviews with individuals knowledgeable in the area.

Recommendations to improve the data collection effort were provided. Among the recommendations was the use of a standard data set and standard collection procedures. The study includes a proposed data collection form containing all items suggested for inclusion in the standard data set.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Lifecycle Service Life, Cost Estimates, Costs, Databases, Software, Product Development	87
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

AFIT Control Number AFIT/GSS/LSY/91D-1

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to determine the potential for current and future applications of AFIT thesis research. Please return completed questionnaires to: AFIT/LSC, Wright-Patterson AFB OH 45433-6583.

1. Did this research contribute to a current research project?

   a. Yes b. No

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not researched it?

   a. Yes b. No

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency received by virtue of AFIT performing the research. Please estimate what this research would have cost in terms of manpower and/or dollars if it had been accomplished under contract or if it had been done in-house.

   Man Years _____ $_____

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3 above), what is your estimate of its significance?

   a. Highly Significant b. Significant c. Slightly Significant d. Of No Significance

5. Comments

_____ _____

Name and Grade Organization

_____ _____

Position or Title Address